

Assignment 2: Designing a Model Context Protocol (MCP) with FastMCP

CS404FZ 2025-2026 S1

1. Assignment Overview

In this assignment, you will design and implement a Model Context Protocol (MCP) server using FastMCP that enables Large Language Models (LLMs) to answer complex, data-driven questions about a real-world dataset.

You are provided with a dataset containing metadata for 20,000 games published on Steam. Rather than querying the dataset directly, the LLM must access the data only through tools exposed by your MCP server, backed by a relational database.

By completing this assignment, you will gain hands-on experience with:

- Data cleaning and preprocessing
- Tool-augmented LLMs via MCP
- Agentic LLMs evaluation and comparison
- Designing agentic AI systems that separate reasoning from data access

2. Dataset

You are given a dataset named `games_sample.csv` (you can download it from **Moodle**), containing structured and semi-structured information about Steam games, including (but not limited to):

- Game metadata (name, release date, price, age requirement)
- Platforms (Windows / macOS / Linux)
- Developers and publishers
- Categories, genres, tags
- User feedback (positive/negative reviews, recommendations)
- Playtime statistics

3. Tasks

Task 1: Dataset Exploration & Database Design (10 marks)

- Explore `games_sample.csv` and identify key entities.
- Design an appropriate MySQL schema (one or more tables).
- Preprocess the data and insert it into the database.
- Justify table structure, keys, and data types.

Task 2: MCP Server Implementation (30 marks)

- Implement an MCP server using Python FastMCP (<https://gofastmcp.com/getting-started/welcome>) .
- Provide multiple tools that query the database.
- Tools must support filtering, aggregation, and comparison.
- Code should be readable and well-documented.

Task 3: MCP Client Setup (10 marks)

- Configure an MCP-compatible client (e.g. Claude Desktop, Cherry Studio).
- Connect it to your MCP server.
- Use at least two different LLMs as the brain model.
- Verify tool invocation works for both models.

Task 4: Dataset Querying via MCP (20 marks)

Interact with MCP-equipped models and ask at least five non-trivial questions for each model that require database access. Include screenshots of conversations showing tool usage and model responses.

Example questions (you may adapt or extend these):

1. How has the average price of Steam games changed over time? Are newer games generally more expensive than older ones?
2. Do games that support Linux receive different user approval (positive vs negative reviews) compared to Windows-only games?
3. Which genres have the highest average playtime, and how does this relate to their average price?
4. Is there a relationship between the number of recommendations and the positive/negative review ratio?
5. Are there publishers whose games consistently achieve higher-than-average user satisfaction?
6. Are large discounts more common in older games, or do newer games also receive deep discounts?

Task 5: Model Comparison & Reflection (25 marks)

- Evaluate outputs from different models.
- Discuss differences in reasoning, accuracy, and clarity.
- Reflect on lessons learned from designing the MCP.

Task 6: Submission & Documentation (5 marks)

Submit a ZIP file named as **assignment2_{your_mu_id}.zip** containing:

- **server.py** : contains the code for the MCP server; you must understand and be able to explain the code you submit.

- **report.pdf**, which should elaborate:
 - Database design and preprocessing
 - MCP server and tool design
 - Client configuration
 - Conversation screenshots and evaluation
 - Reflections and conclusions